100% Pass Unparalleled Amazon - DOP-C02 - Latest AWS Certified DevOps Engineer - Professional Exam Papers

P.S. Free 2026 Amazon DOP-C02 dumps are available on Google Drive shared by Easy4Engine: https://drive.google.com/open?id=13I0VuCnAuOam9JxBXAbbaUJvw_vEZILX

When you prepare for Amazon DOP-C02 certification exam, it is unfavorable to blindly study exam-related knowledge. There is a knack to pass the exam. If you make use of good tools to help you, it not only can save your much more time and also can make you sail through DOP-C02 test with ease. If you want to ask what tool it is, that is, of course Easy4Engine Amazon DOP-C02 exam dumps.

Amazon DOP-C02 exam is a professional-level certification offered by Amazon Web Services (AWS) that validates an individual's expertise in DevOps practices and techniques. DevOps is a rapidly growing approach to software development that emphasizes collaboration, automation, and continuous delivery. AWS Certified DevOps Engineer - Professional certification is designed for experienced DevOps professionals who have a deep understanding of AWS services and DevOps principles.

Amazon DOP-C02 certification is highly valued in the industry, and it is recognized by companies worldwide. It demonstrates the candidate's expertise in designing, deploying, and managing highly available, fault-tolerant, and scalable systems on the AWS platform, and it can open up many career opportunities.

The DOP-C02 Certification Exam consists of 75 multiple-choice and multiple-response questions, which must be completed within 180 minutes. DOP-C02 exam is designed to test the candidate's knowledge across several domains, including Configuration Management and Infrastructure as Code, Monitoring and Logging, Security, Compliance, and Deployment and Provisioning. DOP-C02 exam is computer-based and can be taken at an AWS test center or remotely.

>> Latest DOP-C02 Exam Papers <<

DOP-C02 Reliable Exam Cram, Pdf Demo DOP-C02 Download

Our DOP-C02 Study Materials are convenient for the clients to learn and they save a lot of time and energy for the clients. After the clients pay successfully for the DOP-C02 study materials they can immediately receive our products in the form of mails in 5-10 minutes and then click on the links to use our software to learn. The clients only need 20-30 hours to learn and then they can attend the test. For those in-service office staff and the students who have to focus on their learning this is a good new because they have to commit themselves to the jobs and the learning and don’t have enough time to prepare for the test.

Amazon AWS Certified DevOps Engineer - Professional Sample Questions

(Q146-Q151):

NEW QUESTION # 146
A company has an organization in AWS Organizations. The organization includes workload accounts that contain enterprise applications. The company centrally manages users from an operations account. No users can be created in the workloadaccounts. The company recently added an operations team and must provide the operations team members with administrator access to each workload account.
Which combination of actions will provide this access? (Choose three.)

- A. Create a SysAdmin role in each workload account. Attach the AdministratorAccess policy to the role. Modify the trust relationship to allow the sts:AssumeRole action from the operations account.
- B. In the operations account, create an IAM user group that is named SysAdmins. Add an IAM policy that allows the sts:AssumeRole action for the SysAdmin role in each workload account. Add all operations team members to the group.
- C. Create an Amazon Cognito identity pool in the operations account. Attach the SysAdmin role as an authenticated role.
- D. Create an Amazon Cognito user pool in the operations account. Create an Amazon Cognito user for each operations team member.
- E. In the operations account, create an IAM user for each operations team member.
- F. Create a SysAdmin role in the operations account. Attach the AdministratorAccess policy to the role. Modify the trust relationship to allow the sts:AssumeRole action from the workload accounts.

Answer: A,B,E

Explanation:
https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

NEW QUESTION # 147
A company has multiple AWS accounts. The company uses AWS IAM Identity Center (AWS Single Sign- On) that is integrated with AWS Toolkit for Microsoft Azure DevOps. The attributes for access control feature is enabled in IAM Identity Center.
The attribute mapping list contains two entries. The department key is mapped to ${path:enterprise.department}. The costCenter key is mapped to ${path:enterprise.costCenter}.
All existing Amazon EC2 instances have a department tag that corresponds to three company departments (d1, d2, d3). A DevOps engineer must create policies based on the matching attributes. The policies must minimize administrative effort and must grant each Azure AD user access to only the EC2 instances that are tagged with the user's respective department name.
Which condition key should the DevOps engineer include in the custom permissions policies to meet these requirements?

- A.
```
"Condition": {
    "ForAllValues:StringEquals": {
        "ec2:ResourceTag/department": ["d1", "d2", "d3"]
    }
}
```
- B.
```
"Condition": {
    "StringEquals": {
        "ec2:ResourceTag/department": "$(aws:PrincipalTag/department)"
    }
}
```
- C.
```
"Condition": {
    "ForAllValues:StringEquals": {
        "aws:TagKeys": ["department"]
    }
}
```

- D.

```
"Condition": {
    "StringEquals": {
        "aws:PrincipalTag/department": "${aws:ResourceTag/department}"
    }
}
```


Answer: B

Explanation:
https://docs.aws.amazon.com/singlesignon/latest/userguide/configure-abac.html

NEW QUESTION # 148
A company has its AWS accounts in an organization in AWS Organizations. AWS Config is manually configured in each AWS account. The company needs to implement a solution to centrally configure AWS Config for all accounts in the organization The solution also must record resource changes to a central account.
Which combination of actions should a DevOps engineer perform to meet these requirements? (Choose two.)

- A. Create an AWS Config organization aggregator in the delegated administrator account. Configure data collection from all AWS accounts in the organization and from all AWS Regions.
- B. Configure a delegated administrator account for AWS Config. Enable trusted access for AWS Config in the organization.
- C. Create an AWS CloudFormation template to create an AWS Config aggregator. Configure a CloudFormation stack set to deploy the template to all accounts in the organization.
- D. Configure a delegated administrator account for AWS Config. Create a service-linked role for AWS Config in the organization's management account.
- E. Create an AWS Config organization aggregator in the organization's management account. Configure data collection from all AWS accounts in the organization and from all AWS Regions.

Answer: A,B

Explanation:
https://aws.amazon.com/blogs/mt/org-aggregator-delegated-admin/ https://docs.aws.amazon.com /organizations/latest/userguide/services-that-can-integrate-config.html

NEW QUESTION # 149
A company's development team uses AVMS Cloud Formation to deploy its application resources The team must use for an changes to the environment The team cannot use AWS Management Console or the AWS CLI to make manual changes directly.
The team uses a developer IAM role to access the environment The role is configured with the Administratoraccess managed policy.
The company has created a new Cloudformationdeployment IAM role that has the following policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:*",
                "lambda:*",
                "dynamodb:*"
            ],
            "Resource": "*"
        }
    ]
}
```


The company wants ensure that only CloudFormation can use the new role. The development team cannot make any manual changes to the deployed resources.
Which combination of steps meet these requirements? (Select THREE.)

- A. Configure the IAM to be to get and pass the CloudFormationDeployment role if cloudformation actions for resources,
- B. Update the trust Of the CloudFormationDepoyment role to anow the cloudformation.amazonaws.com AWS principal to perform the iam:AssumeR01e action
- C. Remove me Administratoraccess policy. Assign the ReadOnly/Access managed IAM policy to the developer role Instruct the developers to assume the CloudFormatondeployment role when the developers new stacks
- D. Add an IAM policy to CloudFormationDeplyment to allow cloudformation * on an Add a policy that allows the iam.PassR01e action for ARN of if iam PassedT0Service equal cloudformation.amazonaws.com
- E. Remove the AdministratorAccess policy. Assign the ReadOnlyAccess managed IAM policy to the developer role. Instruct the developers to use the CloudFormationDeployment role as a CloudFormation service role when the developers deploy new stacks.
- F. Update the trust of CloudFormationDeployment role to allow the developer IAM role to assume the CloudFormationDepoyment role.

Answer: B,D,E

Explanation:
A comprehensive and detailed explanation is:
Option A is correct because removing the AdministratorAccess policy and assigning the ReadOnlyAccess managed IAM policy to the developer role is a valid way to prevent the developers from making any manual changes to the deployed resources. The AdministratorAccess policy grants full access to all AWS resources and actions, which is not necessary for the developers. The ReadOnlyAccess policy grants read-only access to most AWS resources and actions, which is sufficient for the developers to view the status of their stacks. Instructing the developers to use the CloudFormationDeployment role as a CloudFormation service role when they deploy new stacks is also a valid way to ensure that only CloudFormation can use the new role. A CloudFormation service role is an IAM role that allows CloudFormation to make calls to resources in a stack on behalf of the user1. The user can specify a service role when they create or update a stack, and CloudFormation will use that role's credentials for all operations that are performed on that stack1.
Option B is incorrect because updating the trust of CloudFormationDeployment role to allow the developer IAM role to assume the CloudFormationDeployment role is not a valid solution. This would allow the developers to manually assume the CloudFormationDeployment role and perform actions on the deployed resources, which is not what the company wants. The trust of CloudFormationDeployment role should only allow the cloudformation.amazonaws.com AWS principal to assume the role, as in option D.
Option C is incorrect because configuring the IAM user to be able to get and pass the CloudFormationDeployment role if cloudformation actions for resources is not a valid solution. This would allow the developers to manually pass the CloudFormationDeployment role to other services or resources, which is not what the company wants. The IAM user should only be able to pass the CloudFormationDeployment role as a service role when they create or update a stack with CloudFormation, as in option A.

Option D is correct because updating the trust of CloudFormationDeployment role to allow the cloudformation.amazonaws.com AWS principal to perform the iam:AssumeRole action is a valid solution. This allows CloudFormation to assume the CloudFormationDeployment role and access resources in other services on behalf of the user2. The trust policy of an IAM role defines which entities can assume the role2. By specifying cloudformation.amazonaws.com as the principal, you grant permission only to CloudFormation to assume this role.
Option E is incorrect because instructing the developers to assume the CloudFormationDeployment role when they deploy new stacks is not a valid solution. This would allow the developers to manually assume the CloudFormationDeployment role and perform actions on the deployed resources, which is not what the company wants. The developers should only use the CloudFormationDeployment role as a service role when they deploy new stacks with CloudFormation, as in option A.
Option F is correct because adding an IAM policy to CloudFormationDeployment that allows cloudformation:* on all resources and adding a policy that allows the iam:PassRole action for ARN of CloudFormationDeployment if iam:PassedToService equals cloudformation.amazonaws.com are valid solutions. The first policy grants permission for CloudFormationDeployment to perform any action with any resource using cloudformation.amazonaws.com as a service principal3. The second policy grants permission for passing this role only if it is passed by cloudformation.amazonaws.com as a service principal4. This ensures that only CloudFormation can use this role.
References:
1: AWS CloudFormation service roles
2: How to use trust policies with IAM roles
3: AWS::IAM::Policy
4: IAM: Pass an IAM role to a specific AWS service

NEW QUESTION # 150
A company uses Amazon EC2 as its primary compute platform. A DevOps team wants to audit the company's EC2 instances to check whether any prohibited applications have been installed on the EC2 instances.
Which solution will meet these requirements with the MOST operational efficiency?

- A. Configure AWS Systems Manager on each instance Use AWS Systems Manager Inventory Use Systems Manager resource data sync to synchronize and store findings in an Amazon S3 bucket Create an AWS Lambda function that runs when new objects are added to the S3 bucket. Configure the Lambda function to identify prohibited applications.
- B. Configure AWS Systems Manager on each instance Use Systems Manager Inventory Create AWS Config rules that monitor changes from Systems Manager Inventory to identify prohibited applications.
- C. Designate Amazon CloudWatch Logs as the log destination for all application instances Run an automated script across all instances to create an inventory of installed applications Configure the script to forward the results to CloudWatch Logs Create a CloudWatch alarm that uses filter patterns to search log data to identify prohibited applications.
- D. Configure AWS Systems Manager on each instance. Use Systems Manager Inventory. Filter a trail in AWS CloudTrail for Systems Manager Inventory events to identify prohibited applications.

Answer: A

Explanation:
Configure AWS Systems Manager on Each Instance:
AWS Systems Manager provides a unified interface for managing AWS resources. Install the Systems Manager agent on each EC2 instance to enable inventory management and other features.
Use AWS Systems Manager Inventory:
Systems Manager Inventory collects metadata about your instances and the software installed on them. This data includes information about applications, network configurations, and more.
Enable Systems Manager Inventory on all EC2 instances to gather detailed information about installed applications.
Use Systems Manager Resource Data Sync to Synchronize and Store Findings in an Amazon S3 Bucket:
Resource Data Sync aggregates inventory data from multiple accounts and regions into a single S3 bucket, making it easier to query and analyze the data.
Configure Resource Data Sync to automatically transfer inventory data to an S3 bucket for centralized storage.
Create an AWS Lambda Function that Runs When New Objects are Added to the S3 Bucket:
Use an S3 event to trigger a Lambda function whenever new inventory data is added to the S3 bucket.
The Lambda function can parse the inventory data and check for the presence of prohibited applications.
Configure the Lambda Function to Identify Prohibited Applications:
The Lambda function should be programmed to scan the inventory data for any known prohibited applications and generate alerts or take appropriate actions if such applications are found.
Example Lambda function in Python
import json
import boto3

def lambda_handler(event, context):
s3 = boto3.client('s3')
bucket = event['Records'][0]['s3']['bucket']['name']
key = event['Records'][0]['s3']['object']['key']
response = s3.get_object(Bucket=bucket, Key=key)
inventory_data = json.loads(response['Body'].read().decode('utf-8'))
prohibited_apps = ['app1', 'app2']
for instance in inventory_data['Instances']:
for app in instance['Applications']:
if app['Name'] in prohibited_apps:
# Send notification or take action
print(f'Prohibited application found: {app['Name']} on instance {instance['InstanceId']}') return {'statusCode': 200, 'body': json.dumps('Check completed')} By leveraging AWS Systems Manager Inventory, Resource Data Sync, and Lambda, this solution provides an efficient and automated way to audit EC2 instances for prohibited applications.
References:
AWS Systems Manager Inventory
AWS Systems Manager Resource Data Sync
S3 Event Notifications
AWS Lambda

NEW QUESTION # 151
.....

What's more, part of that Easy4Engine DOP-C02 dumps now are free: https://drive.google.com/open?id=13I0VuCnAuOam9JxBXAbbaUJvw_vEZILX